

ORACLE (Devam)

```
create user HAFTA13B2 identified by  
"1234"
```

```
default tablespace users  
temporary tablespace temp  
profile default  
quota unlimited on users;
```

```
grant all privileges to HAFTA13B2;
```

```
SELECT ...INTO .... Deyimi
```

Sorgudan elde edilen değerlerin değişkenlere aktarılmasını sağlar. Sorgudan tek satır dönmesi gerekir. Çok satır dönerse hata mesajı verir.

SQL server'da aşağıdaki gibi yapılırdı.

```
SELECT @ortamalama= avg(notu) from  
notlar
```

Oracle için genel ifade;

SELECT kolonİsimleri INTO deęişkenİsimleri
FROM TabloAdi WHERE koşul
şeklindedir.

Örnek:

2053 nolu personelin adını ve soyadını
deęişkenlere aktarınız ve bu deęişkenleri
yazdırınız.

Çözüm:

```
declare
```

```
  p_ad varchar2(20);
```

```
  p_soyad varchar2(20);
```

```
begin
```

```
  select ad, soyad into p_ad, p_soyad from  
  personel where sicilno='2053';
```

```
  dbms_output.put_line(p_ad || ' ' ||  
  p_soyad);
```

end;

yukarıdaki kodu açıklayalım
declare deyimi : değişken tanım bloğudur.

begin...end : çalışacak kodlar yazılır.

p_adi, p_soyadi : değişkenlerdir.

select ad,soyad into p_ad, p_soyad
ile personel tablosundaki ad kolonu değeri
p_adi değişkenine, soyad kolonu değeri
p_soyad değişkenine aktarılır.

dbms_output : çıktıyı sağlayan oracle
paketinin adıdır. Put_line bu paketin
içindeki, yazdırmayı sağlayan prosedürdür.

PL/SQL de atama işlemi:

Atama yapmak için := sembolü kullanılır.

`c := a + b;`

Örnek :

2053 nolu personelin adını ve soyadını isim değişkenine aktarınız.

```
declare
```

```
  p_ad varchar2(20);
```

```
  p_soyad varchar2(20);
```

```
  p_isim varchar2(20);
```

```
begin
```

```
  select ad, soyad into p_ad, p_soyad from
  personel where sicilno='2053';
```

```
  p_isim := p_ad || ' ' || p_soyad;
```

```
  dbms_output.put_line(p_isim);
```

```
end;
```

atama değişken tanımlanırken de yapılabilir.

Örnek

declare

para number(15,2):=2500;

yukarıda para isimli değişkene değer 2500 atanmıştır.

PL/SQL Blokları

- Normal blok
- Fonksiyon bloğu
- Prosedür bloğu

Normal Blok

Az önceki uygulamalar, normal blok yapısına uygun uygulamalardır.

Yapısı:

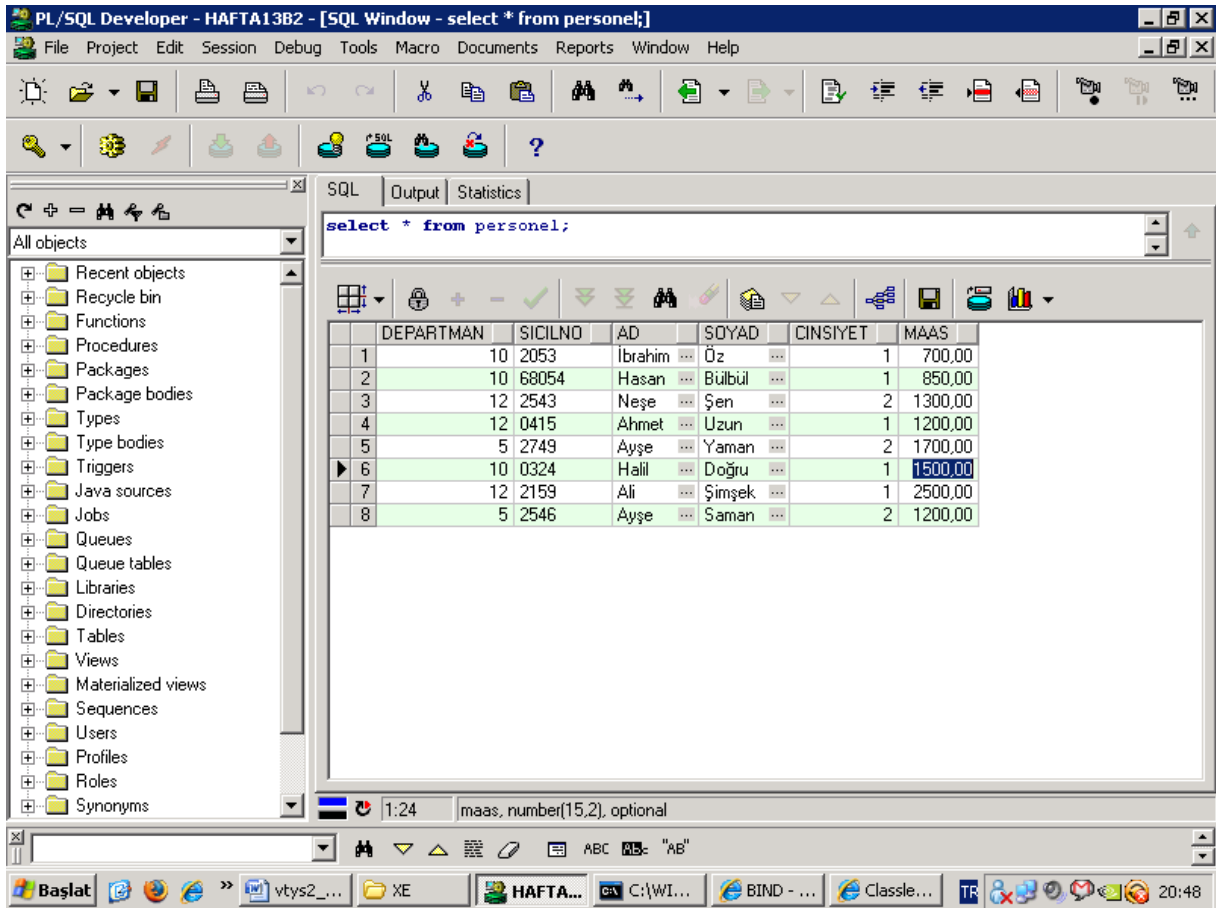
declare

değişken tanımlar

begin

komutlar

EXCEPTION -- hata yakalamak istendiğinde Hata durumunda çalışacak kodlar end;



Örnek:

10 nolu departmanda çalışan personele %15 zam yapan kodu yazınız. Zam oranı değişkeni tanımlayınız ve bu değişkene zam oranını atayınız.

declare

zam_orani number := 0.15;

begin

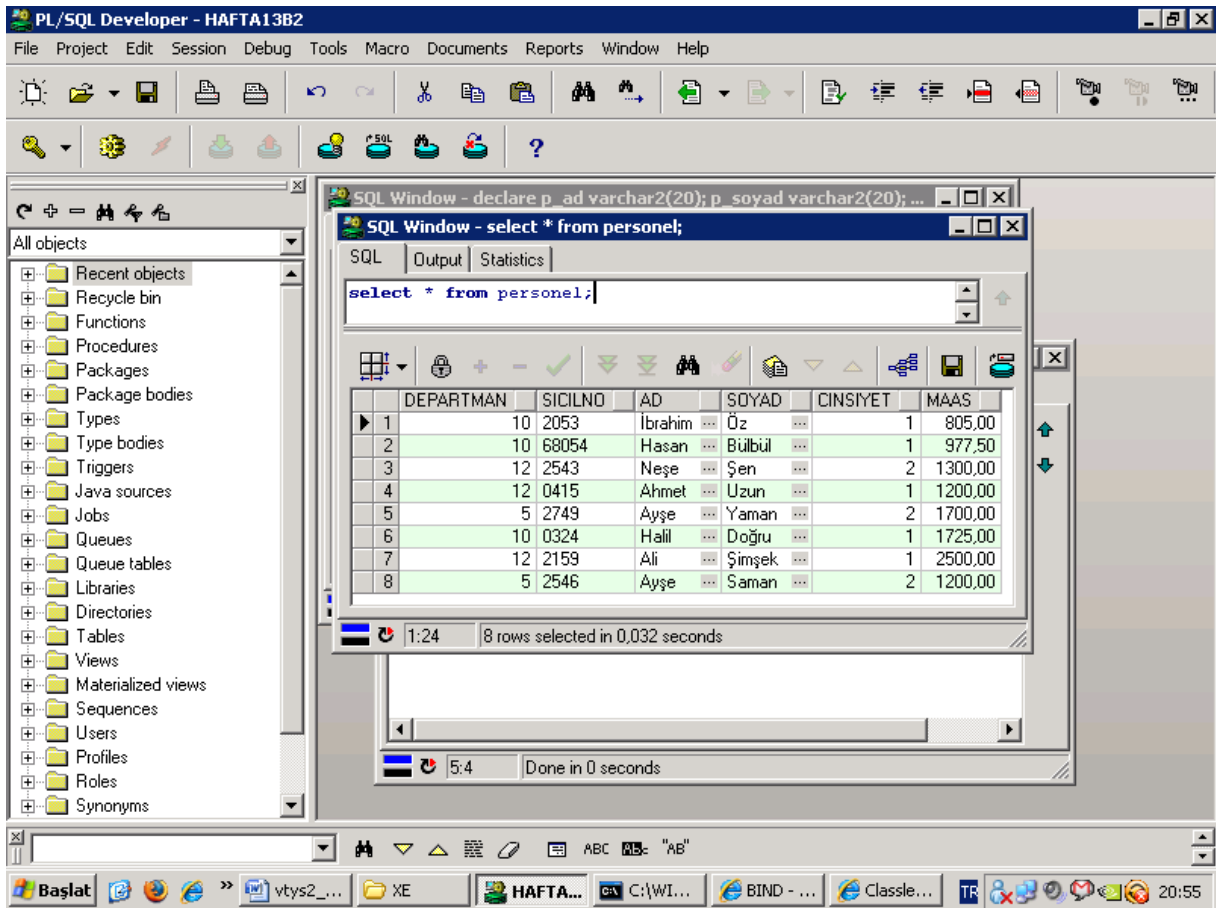
update personel set

maas=maas*(1+zam_orani) where

departman=10;

commit;

end;



Fonksiyon Bloğu

Oracle üzerinde fonksiyon oluşturur. Bu fonksiyonlar scaler değerli fonksiyonlardır. Oracle fonksiyonları tablo döndüremez.

Create or replace function fonksiyonAdi
(parametreler)

return dönüşTipi

IS

 Değişken tanımları

Begin

 Komutlar

EXCEPTION

 Hata durumunda çalışacak kodlar

end;

Örnek:

Sicil numarası verilen personelin maaşını bulan fonksiyon oluşturunuz. Bu fonksiyondan 2053 nolu personelin maaşını bulunuz.

Çözüm:


```
create or replace function
maas_bul(p_sicilno char)
return number
is
  p_maas number;
begin
  select maas into p_maas from personel
where sicilno=p_sicilno;
  return p_maas;
end;
```

şimdi bu fonksiyondan yararlanarak 2053 nolu personelin maaşını bulalım.

```
declare
  maas number;
begin
  maas := maas_bul('2053');

  dbms_output.put_line(maas);
```

end;

```
create or replace function maas_bul(p_sicilno char)
return varchar2
is
  p_maas  number;
  p_ad    varchar2(20);
  p_soyad varchar2(20);
begin
  select maas,ad,soyad into p_maas,p_ad,p_soyad from personel where sicilno=p_sicilno;
  return p_maas||' '||p_ad||' '||p_soyad;
end;
```

```
declare
  maas_isim varchar2(30);
begin
  maas_isim := maas_bul('2053');

  dbms_output.put_line(maas_isim);
end;
```

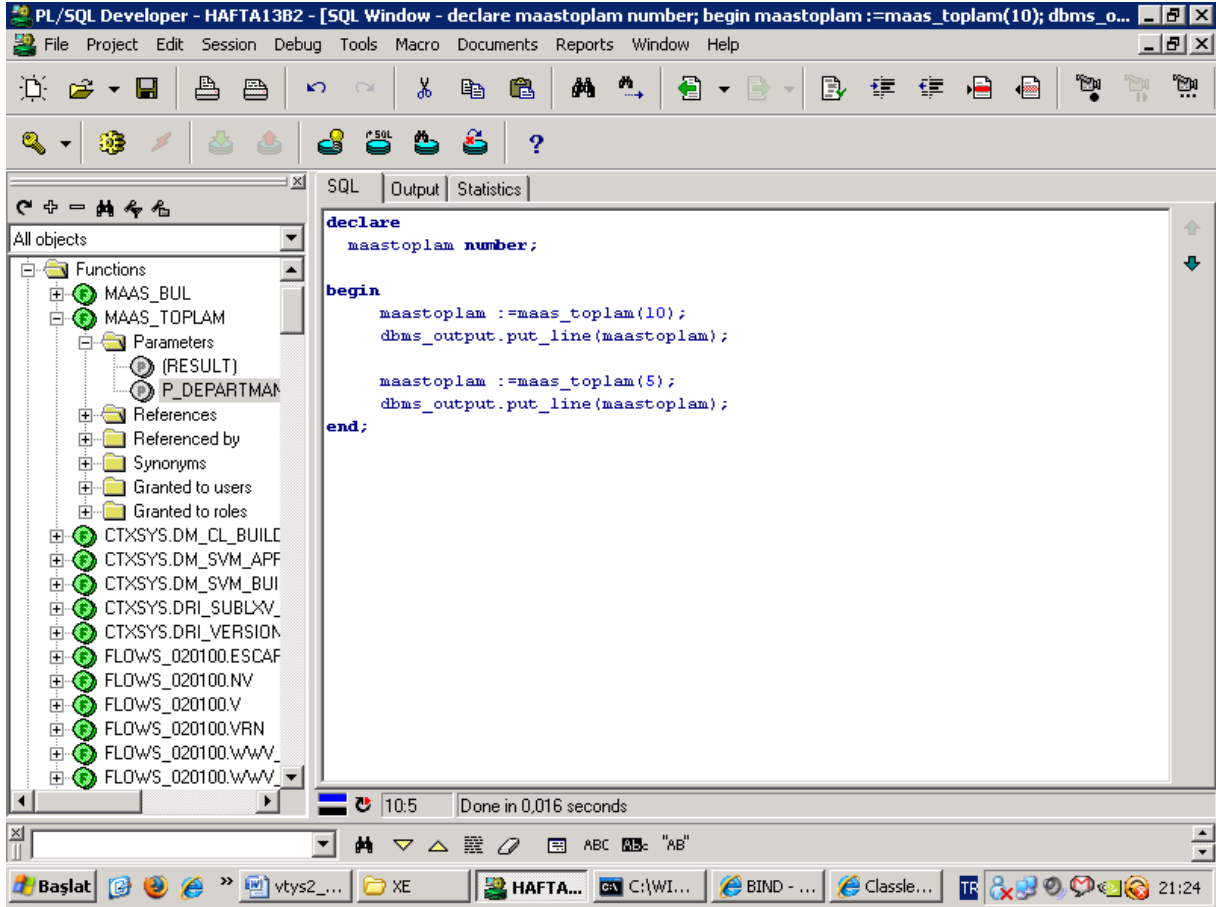
Örnek :

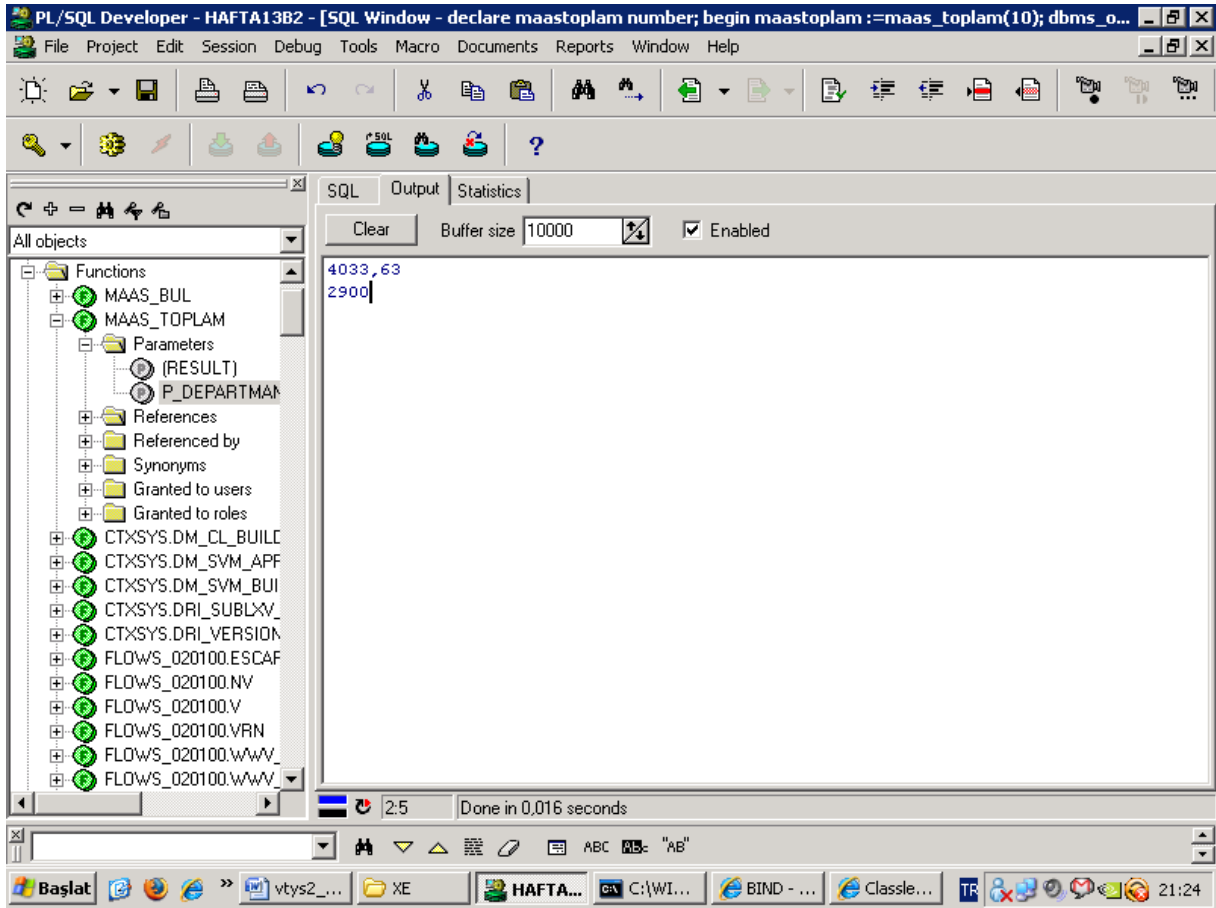
Departman numarası verilen departmanın
Maas toplamını veren fonksiyon
oluşturunuz.

```
create or replace function
maas_toplam(p_departman number)
return number
is
  p_maas_toplam number;
begin
  select sum(maas) into p_maas_toplam
```

```
from personel where departman =  
p_deptman;  
return p_maas_toplam;  
end;
```

yukarıdaki fonksiyonu kullanalım





Prosedür bloğu

Oracle üzerinde prosedür oluşur.

Yapısı:

create or replace procedure prosedürAdi

(parametreler)

IS

 Değişken tanımları

begin

 komutlar

exception

hata durumunda çalışacak kodlar

end;

örnek

personel tablosuna veri gire bir prosedür oluşturunuz.

Çözüm

create or replace procedure

personel_veri_gir

(p_sicilno char, p_ad varchar2, p_soyad
varchar2, p_maas number)

is

begin

insert into personel

(sicilno,ad,soyad,maas)

values(p_sicilno, p_ad,

p_soyad,p_maas);

commit;

end;

bu prosedürü kullanalım.

```
begin
```

```
    personel_veri_gir('4446','Ali','hasan',2500);
```

```
end;
```

Örnek:

Sicil numarası verilen Personelin adını soyadını geri döndüren bir prosedür oluşturunuz.

Prosedürde geri dönüş parametresi out kelimesi ile sağlanır.

Çözüm:

```
create or replace procedure
```

```
personel_isim_bul
```

```
(p_sicilno char, p_adi out varchar2,p_soyadi  
out varchar2)
```

```
is
```

```
begin
```

```
    select ad, soyad into p_adi,p_soyadi from  
personel
```

```
    where sicilno=p_sicilno;
```

end;

Şimdi bu prosedürü kullanarak 2053 nolu personelin adını soyadını öğrenelim.

```
declare
```

```
  p_ad varchar2(20);
```

```
  p_soyad varchar2(20);
```

```
begin
```

```
  personel_isim_bul('2053',p_ad,p_soyad);
```

```
  dbms_output.put_line(p_ad || ' ' ||
```

```
p_soyad);
```

```
  personel_isim_bul(p_sicilno =>
```

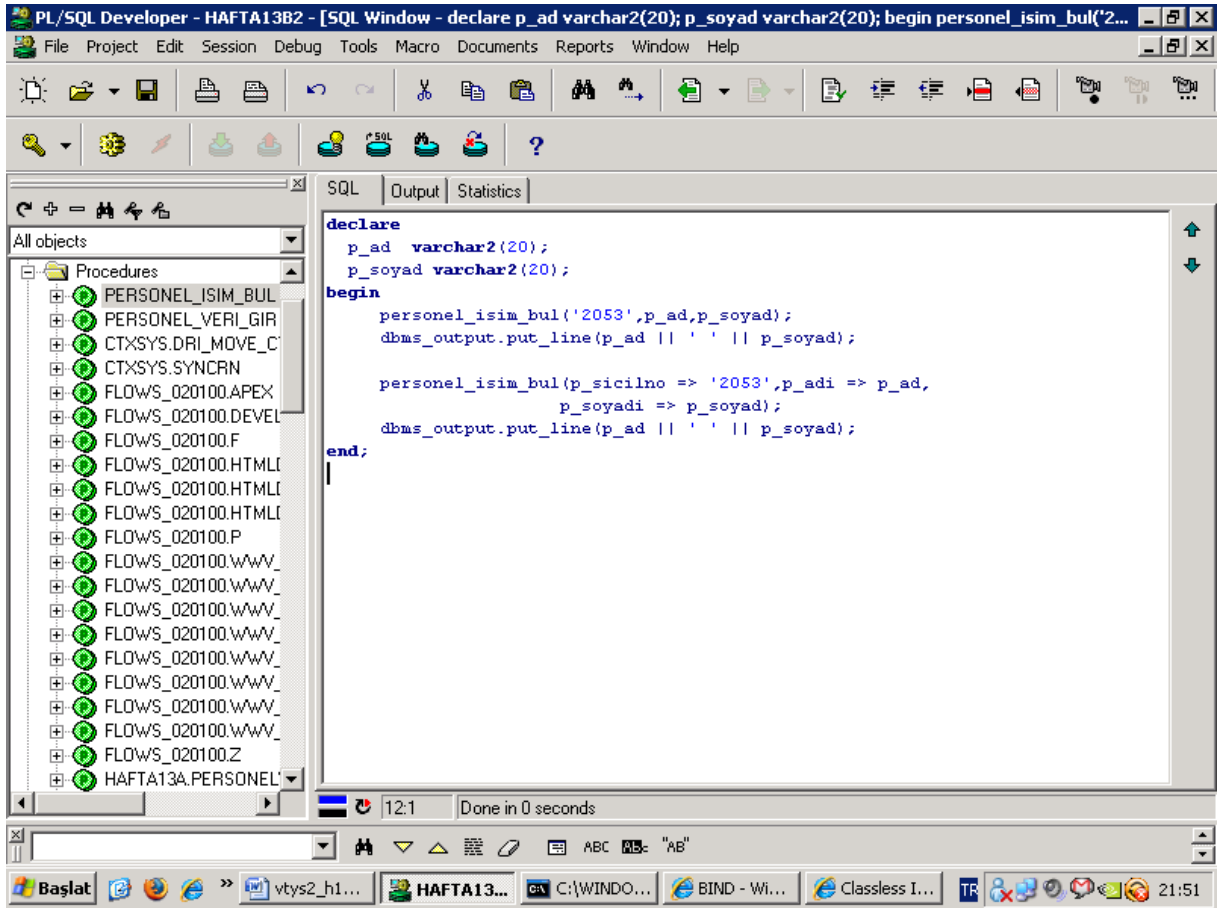
```
'2053',p_adi => p_ad,
```

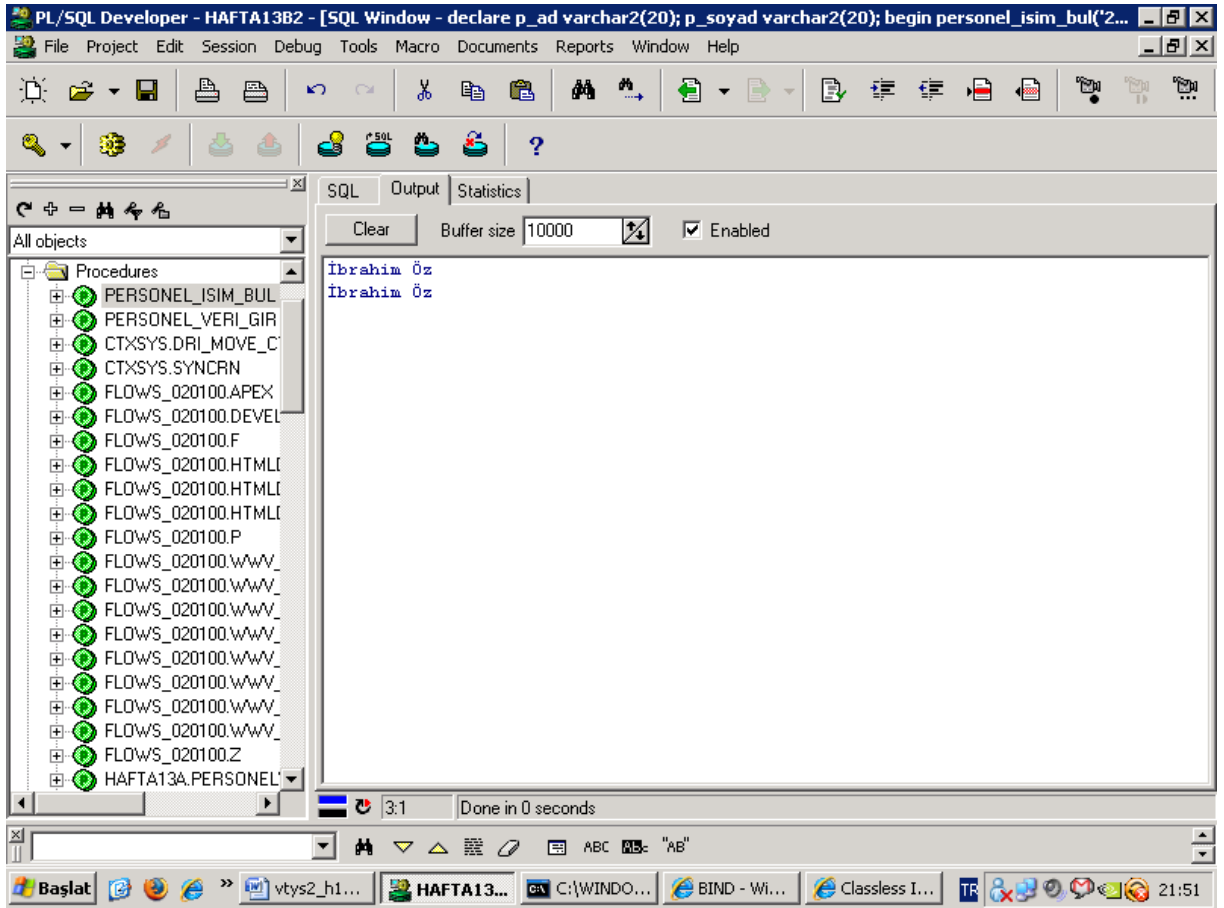
```
    p_soyadi => p_soyad);
```

```
  dbms_output.put_line(p_ad || ' ' ||
```

```
p_soyad);
```

```
end;
```





<http://uygulama.kilicaslan.nom.tr>